Unity Pro 2.2
984 to Unity Converter
User manual

01/2007

33003538.01

a brand of Schneider Electric

Telemecanique

www.telemecanique.com

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

⚡ The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

⚠ This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result** in death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result** in death, serious injury, or equipment damage.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

PLEASE NOTE Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

About the Book

At a Glance

Document Scope

This document describes the functionality and performance scope of the 984 to Unity Converter.

This document is valid for Unity Pro starting from Version 2.2.

Validity Note

The data and illustrations found in this document are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

Related Documents

Title of Documentation	Reference Number
Unity Pro Software Reference Manual	-
Unity Pro Enhanced LL984 Block Library	33003745

You can download these technical publications and other technical information from our website at *www.telemecanique.com*

Product Related Warnings

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this product related warning can result in injury or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techpub@schneider-electric.com

Functional Description

I

Overview

General

This part comprises an overview of the installation and the functional range of the converter.

What's in this Part?

This part contains the following chapters:

Introduction

1

Overview

General

If installed, this converter component is an integrated part of Unity Pro. It is used to convert Modsoft and ProWORX projects and Concept LL984 sections to Unity Pro.

What's in this Chapter?

This chapter contains the following topics:

Overview of the Converter

General

The converter is used to convert Modsoft and ProWORX projects and Concept LL984 sections to Unity Pro.

Coils, contacts, links and function blocks are replaced by the respective objects provided by Unity Pro. Subsequently Unity Pro source code is generated according to the IEC LD format.

Versions

The converter will work with the following versions:

- Modsoft 2.61
- ProWORX V1.11, V1.1, V2.x
- Concept up to V2.6

Starting the Converter

Unity Pro automatically launches the converter after the file to be converted has been selected.

To handle the special formats of Modsoft, ProWORX and Concept LL 984 the converter is embedded in an appropriate framework of the Unity Pro architecture.

By opening the respective project files (*.env, *.pwx, *.asc) the converter is launched out of this framework.

The converter checks whether the project files are valid and show the suitable file formats.

Error Handling

If it is not possible to convert certain objects they will be replaced by dummy EFBs with comments which include detailed error messages.

Error messages will be logged on the report file and will be displayed on the message window during the analysis time. The user is able to go to the error spot of the network by double clicking the error message line in the message window.

Installation and Registration of the Converter

Installation

The converter has to be installed as an additional software component of Unity Pro.

After the installation the converter is an integrated part of Unity Pro.

Registration

Before using the converter has to be registrated.

It is not possible to use the converter in demo mode.

Trying the first time to start the converter, you will be asked for registration.

Converting Process

2

Process of Conversion

General

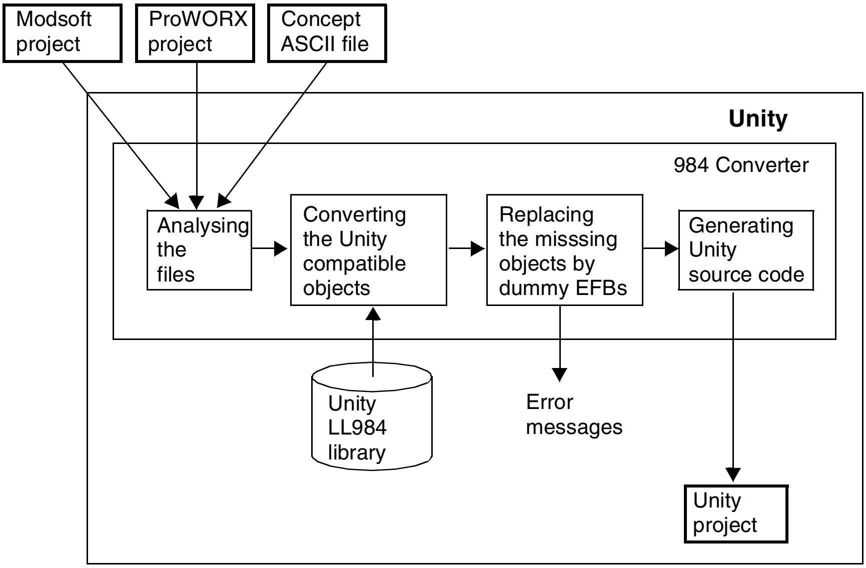
Unity Pro automatically launches the converter after the file to be converted has been selected.

The converter checks the selected file for validity and suitable formats.

Subsequently Unity Pro source code is generated according to the IEC LD format.

Presentation

Presentation of the converting process:



Stages of the conversion:

Stage	Description
1	In Unity Pro the file to be converted (Modsoft-, ProWORX, LL984 Section of an Concept-ASCII-File) is selected. Through this the converter is launched automatically.
2	The selected file is checked for validity and suitable formats by the converter.
3	Coils, contacts, links and function blocks are replaced by the respective objects provided by Unity Pro.
4	If it is not possible to convert certain objects they will be replaced by dummy EFBs with comments which include detailed error messages. Error messages will be logged on the report file and will be displayed on the message window during the analysis time. The user is able to jump to the error spot of the network by double clicking the error message line in the message window.
5	Subsequently Unity Pro source code is generated according to the IEC LD format and is shown in the Unity Pro main window.

Dummy EFBs

Further information you will find under *Dummy EFBs, p. 45*.

Converting Procedure

3

Introduction

Overview

This chapter comprises information on the procedure of conversion.

What's in this Chapter?

This chapter contains the following topics:

Converting a File in Unity Pro

General

Unity Pro automatically launches the converter after the file to be converted has been selected.

The file to be converted can be selected by the menu **File** → **Open**.

Export of a LL984 section out of Concept

LL984 sections in Concept have to be exported into a text file (ASCII file).

After that it is possible to open this file in Unity Pro. The converter will be launched automatically.

Modsoft (*.env) and ProWORX (*.pwx) projects can be opened directly.

Converting a File

Proceed as follows to convert a file to Unity Pro:

Step	Action
1	Open the dialog box for selecting a file by clicking **File** → **Open**.
2	Select the respective file type.
3	Select the source file that should be converted.
4	Confirm with the **Open** button.
5	The converting process will be started and a progress bar is showing the proceeding of the conversion.

Cancel the Conversion

The conversion can be canceled by clicking the **Cancel** button in the progress dialog box.

Trouble Shooting after Converting a File

General

If it is not possible to convert certain objects the converter will replaced them by dummy EFBs with comments which include detailed error messages.

Trouble Shooting

Error messages will be logged on the report file and will be displayed on the message window during the analysis time.

By double clicking the error message line in the message window the user is able to jump to the error spot of the network and correct the errors step by step.

Analyze Project

To ensure that the project contains no more errors after trouble shooting, select the menu command **Build** → **Analyze Project**.

Converted Data

II

Overview

General

This part comprises information on the data that is converted out of Modsoft and ProWORX projects and Concept LL984 sections to Unity Pro.

What's in this Part?

This part contains the following chapters:

Converted Data in General

4

Overview

General

This chapter comprises information on the data that is converted out of Modsoft and ProWORX projects and Concept LL984 sections to Unity Pro and on objects that can not be converted or will be replaced by the respective objects provided by Unity.

What's in this Chapter?

This chapter contains the following topics:

Configuration

Modsoft Configuration Section

Modsoft is a DOS-based PLC programming tool using the LL984 language.

The Modsoft configuration section comprises the following elements:

- PLC name
- Model name (PLC type)
- System memory size
- Extended memory size
- ASCII message table
- I/O port information
- Drop heads
- Segment schedules
- Special functions
- Loadable instructions
- Comments

ProWORX Configuration

ProWORX is unlike Modsoft a Microsoft Windows-based PLC programming tool using the LL984 language.

The ProWORX configuration comprises the following:

- PLC configuration
- Traffic Cop
  - 800 series
  - 200 series
  - DCP series **(not converted)**
  - Momentum **(not converted)**
- Communication
- Configuration Extensions
  - Data Protect
  - Peer Cop
  - S980 Addresses **(not converted)**
  - Quantum Hot Standby
  - Profibus **(not converted)**
  - TCI/IP
  - SY/MAX
  - I/O Scanner
  - Quantum Security **(not converted)**

Note: Please observe that the list items above highlighted as **(not converted)** will **not** be converted from ProWORX to Unity Pro.

LL984 Sections of Concept

A Concept project first has to be exported into a text file (ASCII file).

If such a Concept file contains LL984 sections and the user tries to open that file in Unity Pro, the converter will be launched automatically.

All error messages will be logged on the report file and will be displayed on the message window during the analysis time.By double clicking the error message line in the message window the user is able to jump to the error spot of the network and correct the errors step by step.

Example of a LL984 section in an ASCII file:

```
CP_SEC "_ 1" SECTK_F_SECTION LANG_LL SVB: FALSE ID: 1 CODE_IND:
1 TEXT:
LL_NET 1 NAME: "               " SONDAT: 0 TEXT:
LL_NOD COL,ROW:1,1 OPCOD: 8 VERT:1 REF: 219:_100012 TEXT:
LL_NOD COL,ROW:1,3 OPCOD: 8 VERT:1 REF: 219:_000107 TEXT:
LL_NOD COL,ROW:1,4 OPCOD: 8 VERT:0 REF: 219:_000108 TEXT:
LL_NOD COL,ROW:1,5 OPCOD: 9 VERT:1 REF: 219:_000051 TEXT:
LL_NOD COL,ROW:2,1 OPCOD: 16 VERT:0 REF: 9:15 TEXT:
LL_NOD COL,ROW:2,2 OPCOD: 21 VERT:0 REF: 219:_400001 TEXT:
LL_NOD COL,ROW:2,3 OPCOD: 8 VERT:0 REF: 219:_000051 TEXT:
LL_NOD COL,ROW:2,5 OPCOD: 16 VERT:0 REF: 9:10 TEXT:
LL_NOD COL,ROW:2,6 OPCOD: 21 VERT:0 REF: 219:_400002 TEXT:
LL_NOD COL,ROW:3,5 OPCOD: 12 VERT:0 REF: 219:_000107 TEXT:
LL_NET 2 NAME:"SHUTDOWN ALARM CONTROL" SONDAT:0 TEXT:
LL_NOD COL,ROW:1,1 OPCOD: 9 VERT:1 REF: 219:_000145 TEXT:
LL_NOD COL,ROW:1,5 OPCOD: 9 VERT:1 REF: 219:_000109 TEXT:
LL_NOD COL,ROW:2,1 OPCOD: 17 VERT:0 REF: 219:_400229 TEXT:
LL_REG 0:148 NID:11 COL,ROW: 1,3 LNG: 1 TRUE "OPEN"
EC_UNKNOWN_DT_ID
LL_INS "BLKM" NID: 12 COL,ROW: 1,3
LL_INS "BLKM" NID: 12 COL,ROW: 2,3
LL_INS "BLKM" NID: 12 COL,ROW: 3,3
LL_INS "BLKM" NID: 12 COL,ROW: 5,3
LL_INS "CLSD" NID: 1 COL,ROW: 1,5
LL_INS "CLSD" NID: 2 COL,ROW: 1,5
LL_INS "CLSD" NID: 2 COL,ROW: 1,1
```

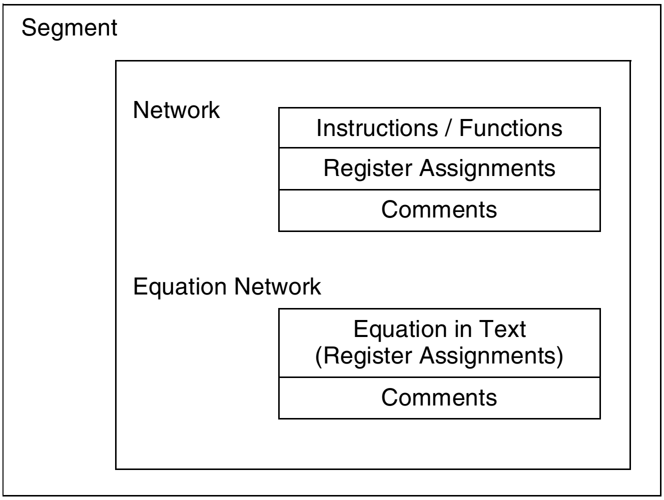
Network Structure

General

Modsoft, ProWORX and the LL984 sections of Concept use the same network structure.

Presentation

Presentation of the network structure:



Constants, Symbols, Networks, Segments and Comments

General

Modsoft, ProWORX and LL984 sections in Concept define constants, symbols, networks, segments and comments.

Formats in Unity Pro

The Modsoft, ProWORX and Concept formats are converted into Unity Pro formats as follows:

Modsoft Format	Unity Pro Format
constant symbol (e.g. #00002)	constant
symbol	variable
symbol descriptor	variable comment
1 or 10 networks	section
network comment	section comment
segment	functional module
segment comment	functional module comment

Note: For each segment (functional module), the first network (section) is associated to segment oriented I/O-transfer (segment scheduling) and the segment comment.

Networks per Section

This option serves for separation of Modsoft, ProWORX and LL984 segments into smaller units.

The separation of segments into smaller units does not take place, if a jump is present in the segment and its target has not yet been reached.

In this case separation becomes active again after the jump target.

Before starting the converting process in Unity Pro the **Networks per Section** option can be set via **Tools** → **Options.**

You can select a value of 1 to 10 networks per section.

Note: To put only 1 network into one section improves the performance of the respective project when editing.

Ignore Jumps When Separating

Before starting the converting process in Unity Pro the **Ignore jumps when separating** option can be set via **Tools** → **Options.**

If these option is set, separation of segments is done regardless of jumps (see also *Networks per Section, p. 28*).

In Unity Pro this will cause analyze errors that have to be resolve manually.

Paging Between Sections

According to the network mapping to sections in Unity Pro, an easy navigation function between sections is established, to provide similar behavior as in ProWORX with its network paging commands.

- Ctrl + Alt + PgUp = display previous network
- Ctrl + Alt + PgDown = display following network

Symbols Instead of Direct Address

This option applies to contacts and coils that have a state RAM address in LL984, which no symbol is defined for.

Before starting the converting process in Unity Pro the **Symbols Instead of Direct Address** option can be set via **Tools** → **Options.**

If it is set, the converter replaces the direct address, which is derived from the state RAM address, by a generated symbol.

Please also refer to *Max. Symbol Length, p. 38*.

State RAM

General

The organization of data memory in the Quantum family is not changed in Unity Pro.

For the Compact and the Momentum families topological addresses are generated.

In order to provide a unified view of data memory, the state RAM areas are mapped using the notation used in IEC 61131. As a second step the data in these areas that correspond to data from I/O modules have an additional, topological address that can be used to access the data.

Notation

The new notation for state RAM maps directly onto the traditional state RAM notation:

Traditional	New	Data Storage
0xxxx	%M or %Q	output coils (bit output)
1xxxx	%I	input coils (bit input)
3xxxx	%IW	input words
4xxxx	%MW or %QW	output words

Example:

Traditional notation of the first output coil:	00001
New notation:	%M1

Forced Outputs (%M)

⚠ **WARNING**

RISK OF UNINTENDED EQUIPMENT OPERATION

Do not rely on memory protect switch.
The behavior of forced outputs (%M) between Modsoft/Proworx/Concept and Unity Pro has changed.

- With Modsoft/ProWORX/Concept you **cannot** force outputs with the Quantum CPU memory protect switch in position "On".
- With Unity Pro you **can** force outputs with the Quantum CPU memory protect switch in position "On".

Failure to follow this instruction can result in death, serious injury, or equipment damage.

⚠ **WARNING**

RISK OF UNINTENDED EQUIPMENT OPERATION

Reforce outputs after a cold start.
The behavior of forced outputs (%M) between Modsoft/Proworx/Concept and Unity Pro has changed.

- With Modsoft/ProWORX/Concept forced outputs **keep** their state after a cold start.
- With Unity Pro forced outputs **lose** their state after a cold start.

Failure to follow this instruction can result in death, serious injury, or equipment damage.

Network Dimensions

LL984 format

The LL984 network dimensions are 11 columns by 7 rows.

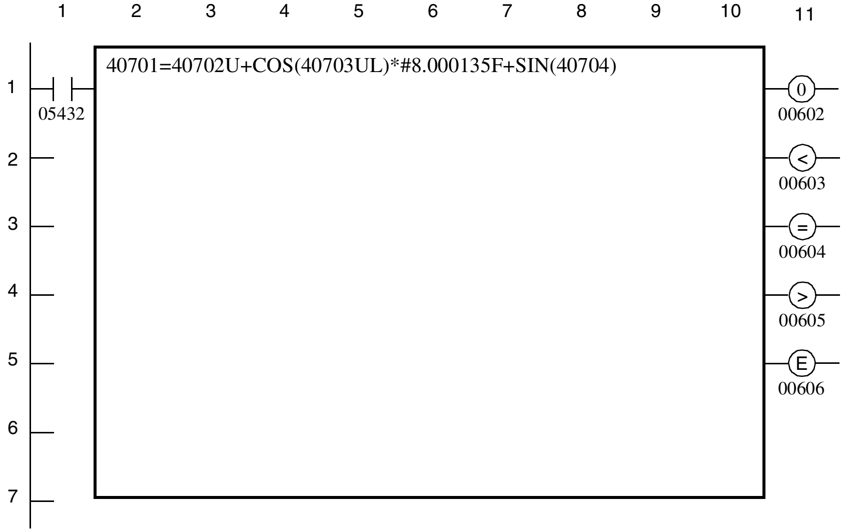
IEC LD format

While converting a LL984 network into an IEC LD network columns and rows are added. Amongst others this is done to create extra space for variables that are shown in additional rows of the function blocks.

Equation Networks ProWORX

ProWORX Equation Network

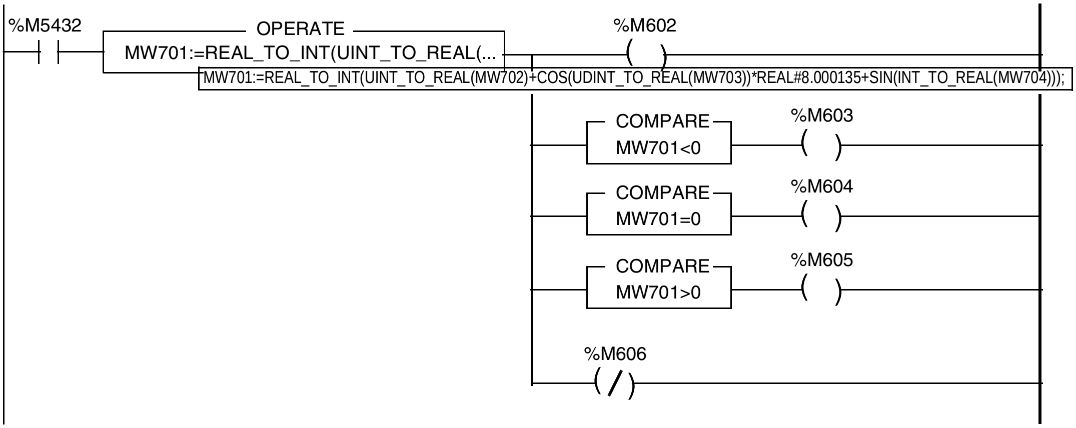
Example of an equation network in ProWORX:



```
40701=40702U+COS(40703UL)*#8.000135F+SIN(40704)
```

Network Converted to Unity Pro

The converted network is shown in Unity Pro as follows:



```
MW701:=REAL_TO_INT(UINT_TO_REAL(MW702)+COS(UDINT_TO_REAL(MW703))*REAL#8.000135+SIN(INT_TO_REAL(MW704)));
```

Not Converted Objects

DX Loadables

DX loadables will be replaced by dummy EFBs with comments which include detailed error messages.

Non-IEC Loadables

IEC incompatible loadables will be replaced by dummy EFBs with comments which include detailed error messages.

IEC incompatible loadables are e.g. `EUCA`, `MBUS`, `PEER` etc.

EXE Loadables

`ULEX` and `XMIT` are EXE loadables. They will not be converted, but will be supported as part of the new high end PLC Unity Executive Runtime.

- `ULEX` is required for Quantum expert I/O modules and is implemented in Unity firmware.
- `XMIT` function block will be supported as an EFB.

ASCII Messages

ASCII messages will not be converted.

6x Range

6x range (register in expanded memory) will not be converted.

Converted Coils, Contacts, Links and Function Blocks

5

Overview

General

This chapter comprises information on the coils, contacts, links and function blocks that are converted out of Modsoft and ProWORX projects and Concept LL984 sections to Unity Pro and on objects that can not be converted or will be replaced by the respective objects provided by Unity Pro.

What's in this Chapter?

This chapter contains the following topics:

Converting Rules and Conversion Settings

Placing Objects and Links

To achieve a proper presentation of the converted LL984 sections in Unity Pro, the following rules are applied.

Rules for object placing:

- The distance between two objects must be at least one cell.
- When two function blocks are connected, the minimum distance must equal the number of cells of the first function block's width.
- The cells in Unity are smaller. If an function block partially occupies another cell, an additional cell is required for the function block.
- If an object (contact or coils) has a vertical link (OR Link), this vertical link will be located at the end of the cell of the object.
- An additional cell is required if:
  - a vertical link (OR Link) with an INPUT function block exists
  - the source function block has output variables
  - the target function block has input variables
- A coil may not be directly connected to the left bus bar.

Rules for the conversion of function block links:

- Function block links between variables/constants and function blocks will be ignored. In these cases, Unity will automatically create a link.
- Purely horizontal function block links between objects that are not function blocks will be replaced with horizontal links with multiple segments.
- When two OR objects are connected, a horizontal link is first connected to the right side of the source OR object. An function block link will then be created between this horizontal link and the target object. This occurs because the two OR objects would otherwise be combined during the import into Unity.
- Each point of the left bus bar can only be occupied by one link.

Suppress Schedule

Sometimes the schedules of Modsoft, ProWORX and Concept I/O drops can not be converted without errors (`Invalid RIO drop`).

Before starting the converting process in Unity Pro the check box **Suppress Schedule** can be set via **Tools** → **Options** to suppress the association of I/O drops (Remote I/O) to sections.

This can be done to avoid analyze errors in Unity Pro.

Suppress Graphs

With Modsoft, ProWORX and Concept comment networks can be edited using links to compose large characters in a graphical way.

Unity Pro does not accept such networks, which contain only links.

Before starting the converting process in Unity Pro the check box **Suppress Graphs** can be set via **Tools** → **Options** to erase all these networks during conversion.

This can be done to avoid analyze errors in Unity Pro.

Attach Coils to Open Ends

Using LL984 programming it is **not** mandatory to terminate a contact network with a coil.

In Unity Pro it **is** mandatory to terminate a contact network with a coil.

Before starting the converting process in Unity Pro the check box **Attach Coils to Open Ends** can be set via **Tools** → **Options**.

If this check box **is set** coils without assigned variables will be positioned at the open ends automatically. This is done to avoid analyze errors in Unity Pro.

Include 984 Execution Corrections

For this option please refer to the chapter *Differences in the Execution Order, p. 51*.

Unpowered LD Objects to Left Power Rail

Using LL984 programming it is allowed to position LD objects (contacts, links) freely in the networks (not connected to the left power rail).

In Unity Pro this would cause analyze errors.

Before starting the converting process in Unity Pro the check box **Unpowered LD Objects to Left Power Rail** can be set via **Tools** → **Options**.

If this check box **is set** unpowered LD objects will be connected to the left power rail automatically.

Blockwidth

The graphical width of blocks can be adjusted to satisfy graphical design needs.

Before starting the converting process in Unity Pro the **Blockwidth** can be set to values from 10 to 15 via **Tools** → **Options**.

Max. Symbol Length

ProWORX LL984 allows to define **non-unique** descriptors to document contact and coil meaning.

Concept LL984 like Unity Pro uses **unique** symbols instead.

The converter constructs unique symbols from ProWORX descriptors by condensing them and appending type and address shortcuts.

Before starting the converting process in Unity Pro the **Max. Symbol Length** can be set via **Tools** → **Options**.

A value of 0 means that no descriptor part is used for the symbol and only the type and address shortcut appears as the unique symbol.

Please also refer to *Symbols Instead of Direct Address, p. 29*.

Objects to recognize transitions

The different ways of handling ladder diagram (LD) objects in Concept (calling an function block) and in Unity Pro (system call) makes the use of State RAM variables (0x/1x register) necessary.

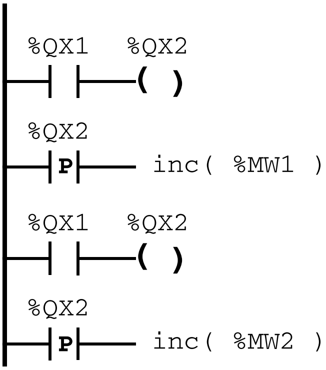
Because of the requirement that several write accesses to the 0x/1x register are possible during a cyclical sweep, there can be differing Online behavior between Concept and Unity Pro.

The objects affected are:
- Positive transition-sensing contact
- Negative transition-sensing contact

In **Concept** the "Old Value" to recognize a transition will only be updated once per cycle.

In **Unity Pro** the "Old Value" will be updated during every write access.

Example:



Concept: Switch `%QX1` from 0 -> 1 and the value of `%MW1` **and** `%MW2` increase.

Unity Pro: Switch `%QX1` from 0 -> 1 and **only** the value of `%MW1` increases.

Note: Use objects to recognize transitions with a certain variable only once per cycle.

Converted Coils, Contacts and Links

General

All coils, contacts and links are converted to Unity Pro technically one-to-one.

Note: Please note the different execution order of transition-sensing objects in Concept and Unity Pro as described under *Objects to recognize transitions, p. 39*.

Converted Coils, Contacts and Links

The following coils, contacts and links are converted to Unity Pro:

- Normal coil
- Normally open contact
- Normally closed contact
- Positive transition-sensing contact
- Negative transition-sensing contact
- Horizontal short
- Vertical short

Note: Retentive coils are converted to normal coils.

Converted Function Blocks

General

All function blocks listed in the tables below are converted to Unity Pro on-to-one.

All function blocks that can **not** be converted you will find under *Not Converted Function Blocks, p. 45*.

Function blocks that can **not** be converted will be replaced by dummy EFBs with comments which include detailed error messages.

Error messages will be logged on the report file and will be displayed on the message window during the analysis time. By double clicking the error message line in the message window the user is able to jump to the error spot of the network and correct the errors step by step.

Dummy EFBs

Further information you will find under *Dummy EFBs, p. 45*.

LL_SENS and LL_MBIT Restriction

Note: The LL_SENS_I and LL_MBIT_I function blocks both have parameters, which either can be auto-incremented variables or constants.
In LL984 the constants cannot be auto-incremented.
The converter translates constants to initialized variables, which will be incremented, if this is configured at the block, while in original LL984 this cannot happen.
You must check all auto-incrementing after conversion.

Converted Function Blocks

The following function blocks are provided by the LL984 library of Unity Pro.

Function blocks provided by the COUNT_TIME family:

LL984 Function Block	Unity EFB (IEC LD)
DCTR	LL_DCTR
T1	LL_T1
T01	LL_T01
T001	LL_T001
T1MS	LL_T1MS
UCTR	LL_UCTR

Function blocks provided by the MATH984LL family:

LL984 Function Block	Unity EFB (IEC LD)
AD16	LL_AD16
ADD	LL_ADD
DV16	LL_DV16
DIV	LL_DIV
MU16	LL_MU16
MUL	LL_MUL
SU16	LL_SU16
SUB	LL_SUB

Function blocks provided by the MATRIX family:

LL984 Function Block	Unity EFB (IEC LD)	Suffixes
AND	LL_AND	_BB, _IB, _BI, _II
BROT	LL_BROT	_BB, _IB, _BI, _II
MBIT	LL_MBIT	_BB, _IB, _BI, _II
NBIT	LL_NBIT	-
NCBT	LL_NCBT	-
NOBT	LL_NOBT	-
OR	LL_OR	_BB, _IB, _BI, _II
RBIT	LL_RBIT	-
SBIT	LL_SBIT	-
SENS	LL_SENS	_X0, _X4
XOR	LL_XOR	_BB, _IB, _BI, _II

Function blocks provided by the MOVE family:

LL984 Function Block	Unity EFB (IEC LD)	Suffixes
BLKM	LL_BLKM	_BB, _IB, _BI, _II
BLKT	LL_BLKT	-
FIN	LL_FIN	_BI, _II
FOUT	LL_FOUT	_BI, _II
R_TO_T	LL_R_TO_T	_BI, _II
T_TO_R	LL_T_TO_R	_BI, _II
T_TO_T	LL_T_TO_T	_BI, _II
TBLK	LL_TBLK	-

A most common example would be the BLKM instruction.

Example 1: To move 40025 through 400124 into 401000 through 401099.

LL_BLKM_II	This block only moves 4x registers into 4x registers.
IN_OFF = 400025	This points to 400025.
OUT_OFF = 401000	This points to 401000.
LENGTH = 100	This asks for 100 registers to be moved.

Example 2: To move 16 0x registers from 000017 through 000034 into 4000102.

LL_BLKM_BI	This block only moves 0x registers into 4x registers.
IN_OFF = 000017	This points to 000017.
OUT_OFF = 400102	This points to 400102
LENGTH = 1	This asks for 16 %M-registers to be moved.

Note: If a register or coil, that is being referred to, does not exist, the function block will fail (with all outputs going off including ENO pin). In this event, the function block will fail only when that non-existent register or coil is encountered.

Suffixes

Meaning of the suffixes:

Suffix	First Parameter	Second Parameter
_BB	0x	0x
_IB	4x	0x
_BI	0x	4x
_II	4x	4x
_B	constant or register	0x
_I	constant or register	4x

Not Converted Function Blocks

General

All function blocks listed in the tables below are **not** converted to Unity Pro one-to-one.

All function blocks that are converted to Unity Pro one-to-one you will find under *Converted Function Blocks, p. 41*.

Function blocks that can not be converted will be replaced by dummy EFBs with comments which include detailed error messages.

Error messages will be logged on the report file and will be displayed on the message window during the analysis time. By double clicking the error message line in the message window the user is able to jump to the error spot of the network and correct the errors step by step.

Dummy EFBs

A dummy EFB provides the following informations:

- The name of a dummy EFB is composed out of "LL_" and the "original name" of the LL984 function block. Example: "LL_INSTRUCTION05".
- All original parameters are listed.
- An error message is displayed in the section:
  `ConvError(Empty DFB to be filled by user)`

Not Converted Function Blocks

The following function blocks are **not** provided by Unity Pro:

Function blocks (ASCII):

LL984 Function Block
READ
WRIT

Function blocks (FAST I/O):

LL984 Function Block
BMDI
ID
IMIO
IE
IMOD
ITMR

Function blocks (MATH):

LL984 Function Block
BCD
TEST
ITOF
FTOI

Function blocks (MATRIX):

LL984 Function Block
CMPR
COMP

Function blocks (MISC):

LL984 Function Block
EMTH
CKSM
MSTR
XMWT
SCIF
XMRD

Function blocks (MOVE):

LL984 Function Block
SRCH
IBKR
IBKW

Function blocks (SKIPS):

LL984 Function Block
SKPC (constant quantity skip)
SKPR (register quantity skip)
LAB
JSR
RET

Function blocks (SPECIAL):

LL984 Function Block
PID2
PCFL
DIOH

Function blocks (LOADABLE):

LL984 Function Block
CHS
EUCA
MBUS
PEER
MAP3
MRTM
DRUM
ICMP
HLTH

Execution Order

III

Overview

General

This part comprises information on the problem that the IEC LD execution order in Unity Pro may differ from the execution order of the original LL984 section in Modsoft, ProWORX or Concept.

What's in this Part?

This part contains the following chapters:

Differences in the Execution Order

6

Overview

General

This chapter comprises information on the problem that the IEC LD execution order in Unity Pro may differ from the execution order of the original LL984 section in Modsoft, ProWORX or Concept.

This fact has to be taken into account while converting a Modsoft or ProWORX project or a Concept LL984 section to Unity Pro.

What's in this Chapter?

This chapter contains the following topics:

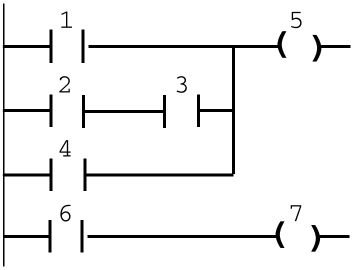
Specific Characteristics

General

The LL984 execution order is different from the execution order in IEC LD.

- The execution order in LL984 is column-by-column.
- The execution order in IEC LD is row-by-row but with the qualification that no element of a network will be evaluated until the states of all of its inputs have been evaluated (according to IEC 61131).

Example of execution order in IEC LD:



Execution Corrections

Before starting the converting process in Unity Pro the check box **Include 984 Execution Corrections** can be set via **Tools** → **Options**.

- If this is done the converter modifies the section so that the result in Unity Pro is the same as in Modsoft, ProWORX or Concept. See *Example:* **Include 984 Execution Corrections**, set, p. 55.
- If the check box is **not** set there may be differences in the result of the respective sections. See *Example:* **Include 984 Execution Corrections**, not set, p. 53.

Error Message

If the check box **Include 984 Execution Corrections** is **not** set, an error message will be logged on the report file and will be displayed on the message window during the analysis time.

This error message reminds the user that the execution order may be changed.

Function Blocks, Execution Order

If the check box **Include 984 Execution Corrections** is set, the converter modifies the section so that the execution order of function blocks in Unity Pro will be the same as in the original LL984 section.

This is done by linking the `EN` and `ENO` parameters of the single function blocks.

Scanning Time

It can not be guaranteed that the scanning time of the converted section is the same as of the original section. May be there will be a longer scanning time.

Example: Include 984 Execution Corrections, not set

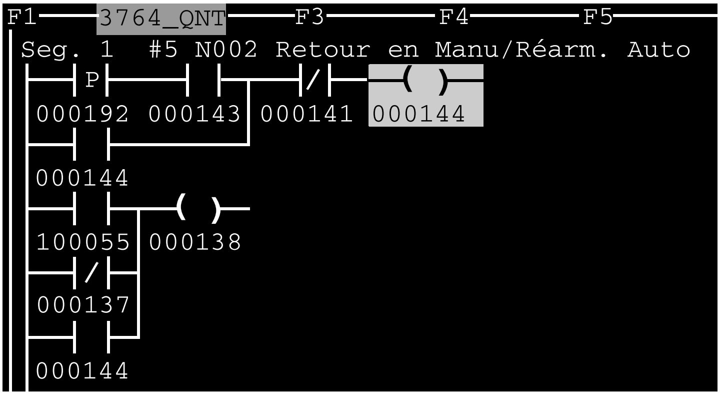
Different Execution Order

If the check box **Include 984 Execution Corrections** is **not** set before starting the converting process in Unity Pro via **Tools** → **Options**, there may be differences in the result of the respective sections.

It may happen that references (coils/contact) are evaluated only when they have changed their state.

LL984 Network

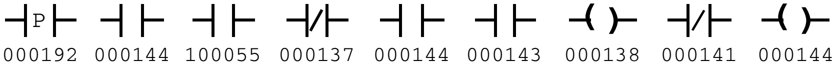
Example network from a LL984 project:



Seg. 1 #5 N002 Retour en Manu/Réarm. Auto

LL984 Execution Order

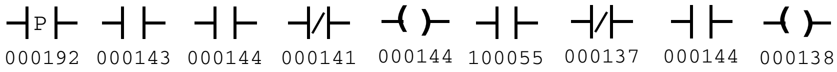
The execution order in LL984 is column-by column:



The normal coil (#000144) is referenced by the normal open contacts in the 2nd and the 5th row and the normal coil (#000144) gets executed at the end. This is the correct order in LL984.

IEC LD Execution Order

The execution order in IEC LD is row by row but with the qualification that no element of a network will be evaluated until the states of all of its inputs have been evaluated:



The normal coil (#000144) gets executed before the referenced normal open contact in the 5th row is evaluated.

Note: The execution order has been changed and the result of the converted IEC LD section will be different from the result of the original LL984 section.

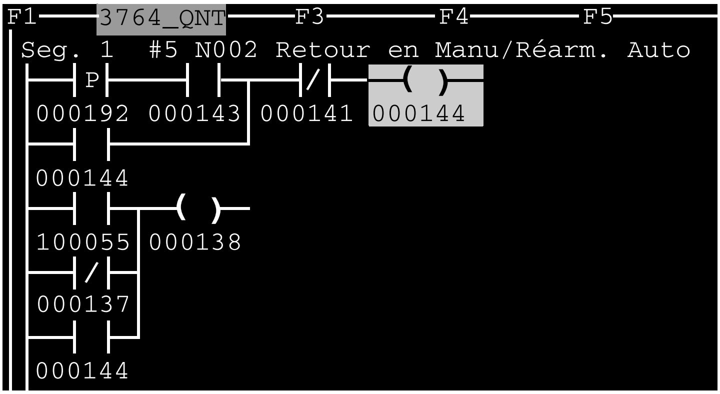
Example: Include 984 Execution Corrections, set

Set the Check Box

If the check box **Include 984 Execution Corrections** is set before starting the converting process in Unity Pro via **Tools** → **Options**, the converter modifies the section so that the result in Unity Pro is the same as in Modsoft, ProWORX or Concept.

LL984 Network

Example network from a LL984 project:

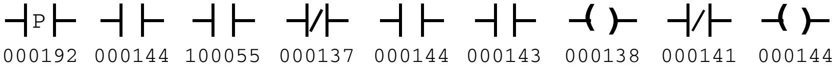


F1 3764_QNT F3 F4 F5

Seg. 1 #5 N002 Retour en Manu/Réarm. Auto

LL984 Execution Order

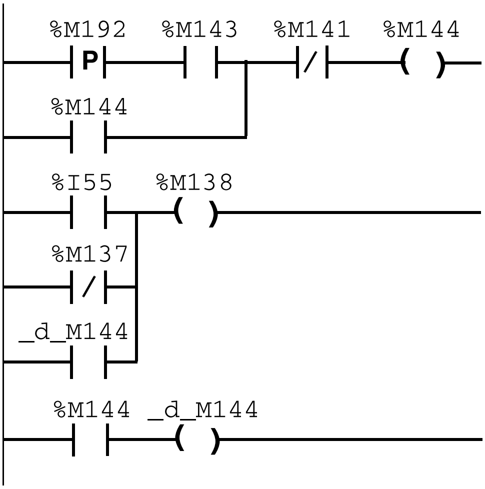
The execution order in LL984 is column-by column:



The normal coil (#000144) is referenced by the normal open contacts in the 2nd and the 5th row and the normal coil (#000144) gets executed at the end. This is the correct order in LL984.

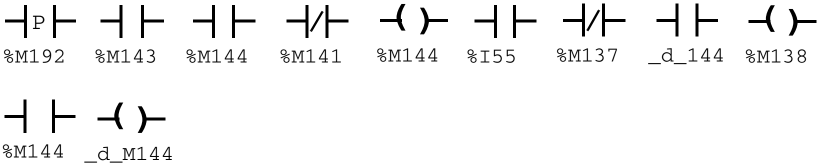
Modified Network

The converter modifies the section by adding auxiliary coils and contacts with the prefix `_d_`:



IEC LD Execution Order

The execution order in IEC LD is now:



So the contact _d_M144 receives the value of the previous execution cycle as it is with the LL984 execution order.

Note: The execution order is maintained but the result of the converted section will be the same as the result of the original LL984 section.

Index